UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/982,020	10/19/2001	Peter Markstein	10008023-1	7131

7590 07/03/2006

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

EXAMINER	
PHAM, CHRYSTINE	

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 07/03/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/982,020	MARKSTEIN ET AL.	
	Examiner	Art Unit	
	Chrystine Pham	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 April 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-18,20-27,29 and 31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1, 3-18, 20-27, 29, and 31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to Amendment filed on April 3, 2006. Claims 1, and 18 have been amended. Claims 2, 19, 28, and 30 have been canceled. Claims 1, 3-18, 20-27, 29, and 31 are presented for examination.

Remarks

2. As an initial matter, regarding the interview dated March 30th, 2006, it should be noted that, at the time of the interview, Mr. Tiep Nguyen was not an attorney of record. Furthermore, Mr. Tiep Nguyen called and requested to have the interview within the next two days. The Examiner suggested that the interview be rescheduled for the following week, however Mr. Nguyen insisted that the interview take place and that it would be a quick interview. In response to Applicants' comment about the Examiner's lack of response following the interview dated March 30th, 2006 (Remarks, page 10, last paragraph), the question/argument raised during the interview by Mr. Nguyen (i.e., whether MacLeod anticipates the claimed register allocation stage that is configured to both generate the intermediate code and allocate the real registers) was not an part of the official record. Hence, the Examiner is of the position that she is not obligated to provide a formal response to an informal request.

Response to Arguments

3. Applicants' arguments filed April 3, 2006 have been fully considered but they are not persuasive.

Regarding claim 18, Applicants essentially contend, "In contrast, MacLeod at FIG.1 and associated text clearly shows a parser 2 generating the intermediate code *prior* to the register allocator 6 allocating the real registers" (Emphasis in original)(Remarks, page 11, last paragraph). Applicants further assert that "The Examiner's contention is illogical and deviates from the clear language of claim 18" (Remarks, page 10, first paragraph). The Examiner strongly and respectfully disagrees.

It should be noted that claim 18 recites "... allocating a plurality of real registers to store a plurality of operands from **said intermediate code** during the step of generating the intermediate code,". It is respectfully submitted that the **real registers allocation step** of claim 18 **requires the operands from said intermediate code**. This clearly implies that said intermediate code **has been generated** (i.e., already-generated). In other words, said intermediate code is already-generated BEFORE the register allocation step **in order to provide the operands which need to be ready and available for real register assignment (i.e., allocation) step**. The language of claim 18 is, thus, illogical because the register allocation step, which requires storing **operands from an already-generated intermediate code**, cannot occur at the same time as the intermediate code is still in the process of **being** generated. Furthermore, contrary to Applicants' argument, step 101 of FIG.1 and step 305 of FIG.3A of Applicants' Disclosure repeatedly reinforce MacLeod's concept of generating the intermediate code from the source code PRIOR to assigning real registers to **operands from the intermediate code**. Without the intermediate

code having been generated (i.e., already-generated) from the source code, no operands can be available for real register assignment (i.e., allocation).

In response to Applicants argument that the claim limitation "generating the intermediate code" recited in lines 4-5 of claim 18 refers back to the limitation "generating intermediate code from a portion of source code" recited in line 2 of claim 18, as established by the Examiner during the interview and as discussed above, the "generating of the intermediate code" of lines 4-5 cannot possibly or logically refer back to the "generating intermediate code directly **from a portion of source code**" (Emphasis added) because it would be illogical and impossible to assign a real register to an operand (in the intermediate code) that is not already available (i.e., intermediate code for that operand has not been completely generated). Thus, in response to Applicants' assertion, "The Examiner contention that MacLeod's rewriting of the already-generated intermediate code to merely substitute the symbolic registers with the allocated real registers as shown in MacLeod at step 24 in FIG.2 reads on the claimed generating the intermediate code is illogical" (Remarks, page 10, first paragraph), it is respectfully submitted that the Examiner's position is being supported by Applicants' Disclosure, specifically at step 101 of FIG.1 & associated text and step 305 of FIG.3A, which clearly disclose generating the intermediate code from the source code PRIOR to the register allocation step. Thus, the limitation "during the step of generating the intermediate code", which is an integral part of "allocating a plurality of real registers to store a plurality of operands from said intermediate code" in lines 3-5 of claim 18 does not refer back to "generating intermediate code from a portion of source code" recited in line 2 of claim 18. Hence, once again, the Examiner respectfully submits that MacLeod's

rewriting of the already-generated intermediate code to merely substitute the symbolic registers with the allocated real registers as shown in MacLeod at step 24 in FIG.2 reads on assigning real registers to intermediate code operands (i.e., "allocating a plurality of real registers to store a plurality of operands from said intermediate code") and thus it reads on the claimed "during the step generating the intermediate code" as recited in lines 3-5 of claim 18.

With respect to claims 1 and 3-17, Applicants argue that "In contrast, MacLeod at FIG.1 and & associated text show an optimizer for optimizing the intermediate code *prior* to the register allocator 6 allocating or selecting the class of real registers" (Remarks, starting on page 12, last paragraph). As has been established in the previous Office Action, Sato was relied upon to teach selecting a class of real registers operable to store said operand (i.e., operable to assign to said operand) (see pages 8-9). Furthermore, it is respectfully submitted that very act of storing said operand (i.e., assigning/allocating a real register from a class of real register) from the intermediate code anticipates the overall process of "optimizing the intermediate code". Since the storing or assigning/allocating a real register to said operand (i.e., optimizing the intermediate code) can only take place AFTER a real register (of a class of real register) has been selected for the storing/assigning/allocating step, McLeod as modified by Sato clearly teach "optimizing the intermediate code subsequent to the step of selecting the class of real registers".

4. In view of the fore going discussion, rejection of claims 1, 3-18, 20-27, 29, and 31 under 35 U.S.C. 102(b) and 103(a) is considered proper and maintained.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claim 18 is rejected under 35 U.S.C. 102(b) as being anticipated by MacLeod (US 6090156, *MacLeod*).

Claim 18

MacLeod teaches a method of (i.e., a compiler configured to) compiling source code comprising steps of:

- generating intermediate code from a portion of source code (see at least *parse 2* FIG.1 & associated text);
- during compiling of the source code, allocating a plurality of real registers (see at least *register allocator 6* FIG.1 & associated text) to store a plurality of operands from said intermediate code during the step of generating the intermediate code (i.e., register allocation stage), wherein the allocating further comprises:

  determining a type of operand for at least one of said plurality of operands (see at least *hardware registers, FREE, INUSE, SYMBOLIC* col.7:8-60; *context spiller 12, FREE hardware registers, SYMBOLIC state* col.8:34-45);

allocating a location in memory for the at least one operand in response to said operand being a particular type of operand (see at least *hardware registers, FREE, INUSE, SYMBOLIC* col.7:8-60; *context spiller 12, FREE hardware registers, SYMBOLIC state* col.8:34-45); and

allocating a real register for said operand (see at least *hardware registers, FREE, INUSE, SYMBOLIC* col.7:8-60; *context spiller 12, FREE hardware registers, SYMBOLIC state* col.8:34-45); and

- o optimizing stage configured to optimize said intermediate code (see at least *optimizer 4* FIG.1 & associated text);
- o a final code stage generating machine-readable code from said intermediate code using said plurality of real registers (see at least *code generator 8* FIG.1 & associated text).

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 1, 3-17, 20-27, 29, and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over MacLeod (US 6090156, *MacLeod*) in view of Sato (US 5261062, *Sato*).

Claim 1

MacLeod teaches a method of allocating registers when compiling source code (see at least *compiler 1, parser 2* FIG.1 & associated text), said method comprising steps of:

- translating source code to intermediate code (see at least *compiler 1, parser 2* FIG.1 & associated text);
- identifying an operand from said intermediate code to store in a real register (see at least *register allocator 6* FIG.1 & associated text; *symbolic registers, intermediate language program, hardware registers* col.6:15-32); and
- during compiling of the source code, selecting at last one subclass of real registers, wherein said at least one subclass includes a register to store said operand (see at least *hardware register, FREE, INUSE, SYMBOLIC* col.7:8-35; FIG.3 & associated text; FIG.4 & associated text).

MacLeod does not expressly disclose said subclass belonging to a selected class [operable to store said operand] (i.e., callee-saved or caller-saved class). However, *Sato* teaches a method of allocating registers when compiling source code (see at least FIGS.1A, 1B, 2A, 2B & associated text) and selecting a class (i.e., callee-saved or caller-saved) of real registers operable to store said operand (see at least *r4-r15, r16-r28* FIG.7 & associated text; col.6:28-col.7:10). *MacLeod* and *Sato* are analogous art because they are directed to register allocation during compilation of source code. It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the teaching of *Sato* into that of *MacLeod* for the inclusion of "selecting a class of real registers operable to store said operand". And the motivation for doing so would

have been to further classify the subclasses of real registers under different classes (i.e., callee-saved and caller-saved) with corresponding costs, i.e., the amount of security of register values between functions in view of the use, thus enabling the compiler to perform register allocation on an extremely saving basis as well as producing optimized code that is able to be executed at high speed (see *Sato r4-r15, r16-r28* FIG.7 & associated text; col.6:28-col.7:10). As discussed above, the act of storing said operand (i.e., assigning/allocating a real register from a class of real register) from the intermediate code anticipates the overall process of "optimizing the intermediate code". Since the storing or assigning/allocating a real register to said operand (i.e., optimizing the intermediate code) can only take place AFTER a real register (of a class of real register) has been selected for the storing/assigning/allocating step, McLeod as modified by Sato clearly teach "optimizing the intermediate code subsequent to the step of selecting the class of real registers".

Claim 3

The rejection of base claim 1 is incorporated. As cited in claim 1, *Sato* teaches wherein said selected class includes one of a callee-saved class (i.e., for storing operands including at least one of local variables, stack items and parameters input by a user) (see at least *r4-r15* FIG.7 & associated text) and a caller-saved class (i.e., for storing operands including a temporary computation) (see *r16-r28* FIG.7 & associated text).

Claim 4

The rejection of base claim 1 is incorporated. *MacLeod* (as modified by *Sato*) further teaches wherein said step of selecting at least one subclass further comprises steps of:

- selecting a first set of subclasses within said selected class (see at least *local context spilling, free registers* col.4:10-65);
- determining whether a register included in said first set of subclasses is available to store said operand (see at least *local context spilling, available hardware registers, symbolic registers, free registers* col.4:10-65); and
- in response to said register being available, storing said operand in said register (see at least *local context spilling, available hardware registers, symbolic registers, free registers* col.4:10-65).

Claim 5

The rejection of base claim 4 is incorporated. *MacLeod* further teaches wherein said first set of subclasses includes at least one of non-used-in-current-operation, non-busy, non-live and non-used subclasses (see at least *hardware registers, FREE, dead hardware registers* col.7:8-60).

Claim 6

The rejection of base claim 4 is incorporated. *Gold* further teaches wherein said step of selecting at least one subclass further comprises steps of:

- selecting a second (or third or fourth) set of subclasses within said selected class in response to said register not being available in said first (or second or third) set of

subclasses (see at least *context spiller 12, FREE hardware registers, SYMBOLIC state* col.8:34-45);

- determining whether a register included in said second (or third or fourth) set of subclasses is available to store said operand (see at least *context spiller 12, FREE hardware registers, SYMBOLIC state* col.8:34-45); and
- in response to said register in said second (or third or fourth) set of subclasses being available, storing said operand in said register in said second (or third or fourth) set of subclasses (see at least *hardware registers, FREE, INUSE, SYMBOLIC* col.7:8-60).

Claim 7

The rejection of base claim 6 is incorporated. *MacLeod* further teaches wherein said second set of subclasses includes at least one of non-used-in-current-operation, non-busy, non-live (see claim 5) and used subclasses (see at least *hardware registers, INUSE, SYMBOLIC* col.7:8-60).

Claim 8

The rejection of base claim 6 is incorporated. Claim recites limitations, which have been addressed in claim 6, therefore, is rejected for the same reasons as cited in claim 6.

Claim 9

The rejection of base claim 8 is incorporated. *MacLeod* further teach wherein said third set of subclasses includes at least one of non-used-in-current-operation, live, and non-busy subclasses (see at least *hardware registers, FREE, INUSE, SYMBOLIC* col.7:8-60).

Claim 10

The rejection of base claim 8 is incorporated. Claim recites limitations, which have been addressed in claim 6, therefore, is rejected for the same reasons as cited in claim 6.

Claim 11

The rejection of base claim 10 is incorporated. *MacLeod* further teach wherein said fourth set of subclasses includes at least one of non-used in current operation and busy subclasses (see at least *hardware registers, FREE, INUSE, SYMBOLIC* col.7:8-60).

Claim 12

The rejection of base claim 11 is incorporated. *MacLeod* further teaches spilling a register in at least one of said busy and said live subclasses prior to storing said operand in said register in at least one of said busy and said live subclasses (see at least col.8:45-55).

Claim 13

The rejection of base claim 11 is incorporated. *MacLeod* further teaches storing said operand in a class (i.e., selected other class) other than selected class in response to a register in said fourth set of subclasses not being available (i.e., selected class of registers not including a

not-used-in-current-operation register) (see at least *hardware registers, FREE, INUSE, SYMBOLIC* col.7:8-60; *context spiller 12, FREE hardware registers, SYMBOLIC state* col.8:34-45).

Claim 14

The rejection of base claim 11 is incorporated. *MacLeod* further teaches marking said register as used-in-current-operation in response to storing said operand in said register (see at least col.8:1-15).

Claim 15

The rejection of base claim 11 is incorporated. *MacLeod* further teaches marking said register storing said operand as live and not-used-in-current-operation in response to translating an instruction of said source code (see at least col.7:8-60).

Claim 16

The rejection of base claim 1 is incorporated. Claim recites limitations, which have been addressed in claim 13, therefore, is rejected for the same reasons as cited in claim 13.

Claim 17

The rejection of base claim 3 is incorporated. Claim recites limitations, which have been addressed in claim 3, therefore, is rejected for the same reasons as cited in claim 3.

Claim 20

The rejection of base claim 18 is incorporated. *MacLeod* does not expressly disclose wherein said particular type of operand includes a local variable. However, *Sato* discloses wherein said particular type of operand includes a local variable (see claim 3).

Claim 21

The rejection of base claim 18 is incorporated. Claim recites limitations, which have been addressed in claims 1, and 3, therefore, is rejected for the same reasons as cited in claims 1, and 3.

Claim 22

The rejection of base claim 21 is incorporated. Claim recites limitations, which have been addressed in claim 3, therefore, is rejected for the same reasons as cited in claim 3.

Claim 23

The rejection of base claim 21 is incorporated. Claim recites limitations, which have been addressed in claim 1, therefore, is rejected for the same reasons as cited in claim 1.

Claim 24

The rejection of base claim 23 is incorporated. Claim recites limitations, which have been addressed in claims 5, 7, 9, 11, therefore, is rejected for the same reasons as cited in claims 5, 7, 9, 11.

Claim 25

Claim recites a compiler version performing the method addressed in claims 1 and 18, therefore, is rejected for the same reasons as cited in claims 1 and 18.

Claim 26

The rejection of base claim 25 is incorporated. Claim recites limitations, which have been addressed in claim 19, therefore, is rejected for the same reasons as cited in claim 19.

Claim 27

The rejection of base claim 26 is incorporated. Claim recites limitations, which have been addressed in claim 20, therefore, is rejected for the same reasons as cited in claim 20.

Claim 29

The rejection of base claim 25 is incorporated. Claim recites limitations, which have been addressed in claim 3, therefore, is rejected for the same reasons as cited in claim 3.

Claim 31

The rejection of base claim 30 is incorporated. Claim recites limitations, which have been addressed in claims 5, 7, 9, 11, therefore, is rejected for the same reasons as cited in claims 5, 7, 9, 11.

Conclusion

9. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chrystine Pham whose telephone number is 571.212.3702. The examiner can normally be reached on Mon-Fri, 8:30am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q Dam can be reached on 571.272.3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

CP
June 25, 2006

TUAN DAM
SUPERVISORY PATENT EXAMINER